**AMENDMENT TO THE CLAIMS:**

This listing of claims will replace all prior versions, and listings of claims in the application:

I claim:

1. (Cancel)

2. (Cancel)

3. (Cancel)

4. (Currently Amended) ~~The iterative decoding method of claim 2,~~ A method of decoding for parallel-concatenated convolutional codes that consist of at least two binary convolutional constituent subcodes of finite blocklength value and share a block of information-bits with respect to corresponding interleaver-orderings, so as to produce a block of said blocklength value decoded binary bits for the information-bits that are an approximation to decoded bits obtained from an iterative maximum a posteriori decoder when initially given code-bit channel-symbol values or quantized digital data representing channel-symbol values, wherein the decoding method is a procedure comprising the steps:

(a) quantizing received code-bit channel-symbol values into digital data,

(b) initializing by storing of digital data corresponding to code-bits into assigned memory locations as well as storing appropriate initialized digital data values at required locations including the memory locations that are shared by constituent subcodes to represent maximum a posteriori reliability estimates which are initialized to digital data representing equally likely estimates as also are the memory locations for extrinsic estimates and punctured code-bits,

(c) applying of a recursive table-lookup decoding method for each constituent convolutional subcode while utilizing a set of pre-stored lookup tables and memory locations for each subcode,

(d) applying more iterations of step (c) until a total of iterations has been completed,

(e) extracting decoded binary estimates for the information-bits from the digital data at the shared memory locations representing the maximum a posteriori estimates by utilizing a most significant bit of the digital data, wherein the recursive table-lookup decoding method is a decoding method for a binary convolutional code of said finite blocklength value~~Z~~ so as to produce:

memory updated with digital data representing approximations for the maximum a posteriori reliability estimates for the number, equal to blocklength value, of~~Z~~ information-bits;

and memory updated with digital data representing approximations for the extrinsic reliability estimates for the number, equal to blocklength value of~~Z~~ information-bits,

wherein the decoding method is a recursive table-lookup procedure comprising the steps:

(a) ~~initialization~~initializing by ~~writing/~~storing of data into memory,

(b) reading data from the memory,

(c) reading data from a set of pre-stored lookup tables,

(d) ~~writing/~~storing data read from lookup tables into the memory,

(e) incrementing a memory location-address pointer,

when initially given memory stored with digital data representing:

the channel-symbol reliability estimates for the number, equal to blocklength value, of~~Z~~ parity-bits;

previous extrinsic reliability estimates for the number, equal to blocklength value, of~~Z~~ information-bits;

and maximum a posteriori reliability estimates for the number, equal to blocklength value, of~~Z~~ information-bits, which is a function of the previous extrinsic estimates, ~~the~~ a priori estimates, and the channel-symbol estimates.

5. (Currently Amended) The ~~recursive table-lookup~~ decoding method of claim 4, wherein the locations~~/entries~~ of the data to be read out from lookup tables at some current recursion are determined from at least one of:

data read from the lookup tables in the current recursion; ~~and/or~~

data read from the lookup tables in previous recursions; ~~and/or~~

data read from memory which was stored previously when read from ~~a~~one of the lookup tables in a previous recursion; and~~/or,~~

data read from the memory which was initially stored.

6. (Currently Amended) The ~~recursive table-lookup~~ decoding method of claim 5, wherein the locations~~/entries~~ of the data to be read out from the lookup tables are digital address-words that are formed by appending together one ~~of~~or more digital data-words which are read out from the lookup tables and~~/or~~ memory.

7. (Currently Amended) The ~~recursive table-lookup~~ decoding method of claim 4, wherein ~~the~~a number of recursions is twice the number, equal to blocklength value, ~~blocksize, Z,~~ of the convolutional code and the locations~~/entries~~ of the data to be read out from memory during a recursion are digital address-words that for the first number, equal to blocklength value, of ~~Z~~ recursions will increment sequentially from an address-word value of zero to an address-word value of ~~(Z minus one)~~ the number, equal to blocklength value, minus one and then for the second number, equal to blocklength value, of~~Z~~ recursions will increment by decreasing sequentially from (the number, equal to blocklength value,~~Z~~ minus one) to zero.

8. (Currently Amended) The ~~recursive table-lookup~~ decoding method of claim 4, wherein the number of recursions is twice the number, equal to blocklength value, ~~blocksize, Z,~~ of the convolutional code and the locations~~/entries~~ of the data to be read out from memory during a recursion are digital address-words that for the first number, equal to blocklength value, of~~Z~~ recursions will increment with respect to a permuted ordering of the digital address-word values of zero to (the number, equal to blocklength value,~~Z~~ minus one) and then for the second number, equal to blocklength value, of~~Z~~ recursions will increment through ~~the~~a reverse of ~~the~~a permuted ordering.

9. (Currently Amended) The ~~recursive table-lookup~~ decoding method of claim 4, wherein ~~the~~<u>a</u> number of ~~seperate~~ <u>separate</u> lookup tables in ~~the~~<u>a</u> set of lookup tables is a design parameter where ~~seperate~~ <u>separate</u> lookup tables can be combined to form fewer lookup tables, or ~~seperate~~ <u>separate</u> lookup tables can be split into several lookup tables.

10. (Currently Amended) The ~~recursive table-lookup~~ decoding method of claim 4, wherein the digital data-words that are pre-stored in~~to~~ the lookup tables is a design parameter, where ~~the~~<u>a</u> best mode of operation selects pre-stored data values based on selected inherent ~~mathematical/~~computational functions and selected inherent quantization functions that the ~~lookup-tables~~<u>lookup tables</u> are approximating.

11. (Currently Amended) The ~~recursive table-lookup~~ decoding method of claim 4, wherein ~~the~~<u>a</u> set of lookup tables are pre-stored with digital data-words, based on inherent ~~mathematical/~~computational functions and quantization functions such that the produced decoded data-words representing an approximation to the maximum a posteriori estimate are approximating a modified version of the maximum a posteriori estimate, including the modification that adds a sensitivity factor to ~~the~~ forward state probabilities and ~~the~~ reverse state probabilities within the inherent functions.

12. (Cancel)

13. (Currently Amended) <u>A</u>~~The~~ <u>hardware implemented</u> recursive table-lookup decoding method <u>comprising a recursive procedure of the steps:</u>

<u>(a) initializing by storing of data into memory contained in said hardware,</u>

<u>(b) reading data from said memory,</u>

<u>(c) reading data from a set of pre-stored lookup tables,</u>

<u>(d) storing data read from lookup tables into memory,</u>

<u>(e) incrementing a memory location-address pointer,</u>

<u>for the decoding of binary convolutional codes of fixed blocklength such as to produce a block of estimates which approximate maximum a posteriori estimates for information-bits, and to produce a block of estimates which approximate extrinsic estimates for the</u>

information-bits, ~~of claim 12,~~ wherein a binary convolutional code of finite number, equal to a blocklength value, ~~blocklength Z~~ is being decoded so as to produce:

memory updated with digital data representing approximations for the maximum a posteriori reliability estimates for the number, equal to a blocklength value, of~~Z~~ information-bits;

and memory updated with digital data representing approximations for the extrinsic reliability estimates for the number, equal to a blocklength value, of~~Z~~ information-bits,

when initially given memory stored with digital data representing functions of:

~~the~~ channel-symbol reliability estimates for the number, equal to a blocklength value, of~~Z~~ parity-bits and the number, equal to a blocklength value, of~~Z~~ information-bits; and ~~the~~ a priori reliability estimates for the number, equal to a blocklength value, of~~Z~~ information-bits.

14. (Currently Amended) The hardware implemented recursive table-lookup decoding method of claim 13, wherein the locations~~/entries~~ of the data to be read out from lookup tables at some current recursion are determined from at least one of:

data read from lookup tables in the current recursion; ~~and/or~~

data read from lookup tables in previous recursions; ~~and/or~~

data read from memory which was stored previously when read from a lookup table in a previous recursion; and~~/or~~,

data read from memory which was initially stored.

15. (Currently Amended) The hardware implemented recursive table-lookup decoding method of claim 14, wherein the locations~~/entries~~ of the data to be read out from lookup tables are digital address-words that are formed by appending together one ~~of~~or more digital data-words which are read out from lookup tables and~~/or~~ memory.

16. (Currently Amended) The hardware implemented recursive table-lookup decoding method of claim 13, wherein the given data representing functions of the

reliability estimates are given as: ~~the~~ channel-symbol reliability estimates for ~~the~~ parity-bits; some given appropriate estimates for the information-bits;

and ~~reliabiltiy~~reliability estimates that are a combination of the given appropriate estimates for the information-bits, the a priori estimates for the information-bits, and ~~the~~ channel-symbol estimates for the information-bits.

17. (Currently Amended) The hardware implemented recursive table-lookup decoding method of claim 13, wherein the number of recursions is twice the number, equal to a blocklength value,~~blocksize, Z,~~ of the convolutional code and the locations/~~entries~~ of the data to be read out from memory during a recursion are digital address-words that for the first number, equal to a blocklength value, of~~Z~~ recursions will increment sequentially from an address-word value of zero to an address-word value of (number, equal to a blocklength value~~Z~~ minus one) and then for the second number, equal to a blocklength value, of~~Z~~ recursions will increment by decreasing sequentially from (number, equal to a blocklength value~~Z~~ minus one) to zero.

18. (Currently Amended) The hardware implemented recursive table-lookup decoding method of claim 13, wherein the number of recursions is twice the number, equal to a blocklength value, ~~blocksize, Z,~~ of the convolutional code and the locations/~~entries~~ of the data to be read out from memory during a recursion are digital address-words that for the first number, equal to blocklength value, of~~Z~~ recursions will increment with respect to a permuted ordering of the digital address-word values of zero to (number, equal to blocklength value,~~Z~~ minus one) and then for the second number, equal to blocklength value, of~~Z~~ recursions will increment through ~~the~~a reverse of ~~the~~a permuted ordering.

19. (Currently Amended) The hardware implemented recursive table-lookup decoding method of claim 13, wherein the number of ~~seperate~~separate lookup tables in ~~the~~a set of lookup tables is a design parameter where ~~seperate~~separate lookup tables can be combined to form fewer lookup tables, or ~~seperate~~separate lookup tables can be split into several lookup tables.

20. (Currently Amended) The hardware implemented recursive table-lookup decoding method of claim 13, wherein the digital data-words that are pre-stored ~~into~~ the lookup tables is a design parameter, where ~~the~~a best mode of operation selects pre-stored data values based on selected inherent ~~mathematical/~~computational functions and selected inherent quantization functions that the ~~lookup tables~~lookup tables are approximating.

21. (Currently Amended) The hardware implanted recursive table-lookup decoding method of claim 13, wherein the set of lookup tables are pre-stored with digital data-words, based on inherent ~~mathematical/~~computational functions and quantization functions such that the produced decoded data-words representing an approximation to the maximum a posteriori estimate are approximating a modified version of the maximum a posteriori estimate, including the modification that adds a sensitivity factor to ~~the~~ forward state probabilities and ~~the~~ reverse state probabilities within ~~the~~ inherent functions.